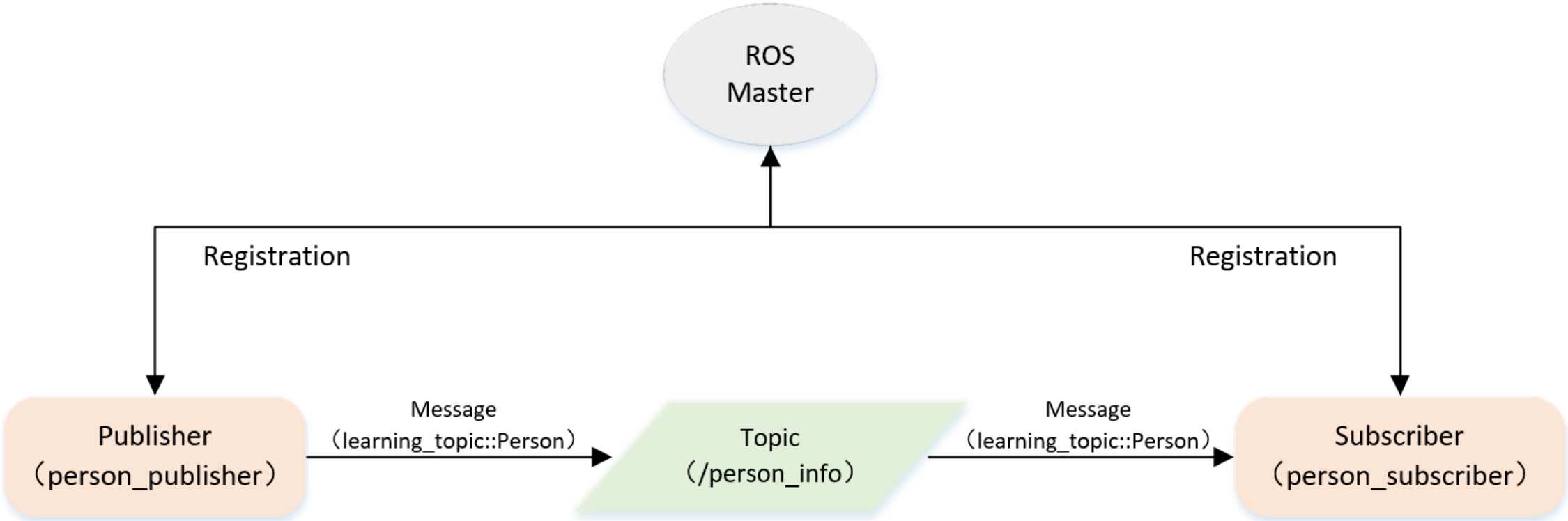
古月居 GYH.AI ROS

ROS入门
21讲

12.话题消息的定义与使用

主讲人：古月

话题模型



话题模型（发布/订阅）

古月居 GYH.AI

如何自定义话题消息

```
string name
uint8  sex
uint8  age

uint8 unknown = 0
uint8 male    = 1
uint8 female  = 2
```

Person.msg

- **定义msg文件；**
- **在package.xml中添加功能包依赖**

```
<build_depend>message_generation</build_depend>
<exec_depend>message_runtime</exec_depend>
```

- **在CMakeLists.txt添加编译选项**
  - find_package(…… message_generation)
  - add_message_files(FILES Person.msg)
    generate_messages(DEPENDENCIES std_msgs)
  - catkin_package(…… message_runtime)
- **编译生成语言相关文件**

创建发布者代码（C++）

```cpp
/**
 * 该例程将发布/person_info话题，自定义消息类型learning_topic::Person
 */

#include <ros/ros.h>
#include "learning_topic/Person.h"

int main(int argc, char **argv)
{
    // ROS节点初始化
    ros::init(argc, argv, "person_publisher");

    // 创建节点句柄
    ros::NodeHandle n;

    // 创建一个Publisher，发布名为/person_info的topic，消息类型为learning_topic::Person，队列长度10
    ros::Publisher person_info_pub = n.advertise<learning_topic::Person>("/person_info", 10);

    // 设置循环的频率
    ros::Rate loop_rate(1);

    int count = 0;
    while (ros::ok())
    {
        // 初始化learning_topic::Person类型的消息
        learning_topic::Person person_msg;
        person_msg.name = "Tom";
        person_msg.age  = 18;
        person_msg.sex  = learning_topic::Person::male;

        // 发布消息
        person_info_pub.publish(person_msg);

        ROS_INFO("Publish Person Info: name:%s  age:%d  sex:%d",
                 person_msg.name.c_str(), person_msg.age, person_msg.sex);

        // 按照循环频率延时
        loop_rate.sleep();
    }
```

person_publisher.cpp

如何实现一个发布者

- 初始化ROS节点；
- 向ROS Master注册节点信息，包括发布的话题名和话题中的消息类型；
- 创建消息数据；
- 按照一定频率循环发布消息。

创建订阅者代码（C++）

```cpp
/**
 * 该例程将订阅/person_info话题，自定义消息类型learning_topic::Person
 */

#include <ros/ros.h>
#include "learning_topic/Person.h"

// 接收到订阅的消息后，会进入消息回调函数
void personInfoCallback(const learning_topic::Person::ConstPtr& msg)
{
    // 将接收到的消息打印出来
    ROS_INFO("Subcribe Person Info: name:%s  age:%d  sex:%d",
             msg->name.c_str(), msg->age, msg->sex);
}

int main(int argc, char **argv)
{
    // 初始化ROS节点
    ros::init(argc, argv, "person_subscriber");

    // 创建节点句柄
    ros::NodeHandle n;

    // 创建一个Subscriber，订阅名为/person_info的topic，注册回调函数personInfoCallback
    ros::Subscriber person_info_sub = n.subscribe("/person_info", 10, personInfoCallback);

    // 循环等待回调函数
    ros::spin();

    return 0;
}
```

person_subscriber.cpp

如何实现一个订阅者

- 初始化ROS节点；
- 订阅需要的话题；
- 循环等待话题消息，接收到消息后进入回调函数；
- 在回调函数中完成消息处理。

配置代码编译规则

```
## Declare a C++ executable
## With catkin_make all packages are built within a single CMake context
## The recommended prefix ensures that target names across packages don't collide
# add_executable(${PROJECT_NAME}_node src/learning_topic_node.cpp)

## Specify libraries to link a library or executable target against
# target_link_libraries(${PROJECT_NAME}_node
#   ${catkin_LIBRARIES}
# )

## Add cmake target dependencies of the executable
## same as for the library above
# add_dependencies(${PROJECT_NAME}_node ${${PROJECT_NAME}_EXPORTED_TARGETS} ${catkin_EXPORTED_TARGETS})

add_executable(person_publisher src/person_publisher.cpp)
target_link_libraries(person_publisher ${catkin_LIBRARIES})
add_dependencies(person_publisher ${PROJECT_NAME}_generate_messages_cpp)

add_executable(person_subscriber src/person_subscriber.cpp)
target_link_libraries(person_subscriber ${catkin_LIBRARIES})
add_dependencies(person_subscriber ${PROJECT_NAME}_generate_messages_cpp)
```

如何配置CMakeLists.txt中的编译规则

- 设置需要编译的代码和生成的可执行文件；
- 设置链接库；
- 添加依赖项。

```
add_executable(person_publisher src/person_publisher.cpp)
target_link_libraries(person_publisher ${catkin_LIBRARIES})
add_dependencies(person_publisher ${PROJECT_NAME}_generate_messages_cpp)

add_executable(person_subscriber src/person_subscriber.cpp)
target_link_libraries(person_subscriber ${catkin_LIBRARIES})
add_dependencies(person_subscriber ${PROJECT_NAME}_generate_messages_cpp)
```

CMakeLists.txt

编译并运行发布者和订阅者

```
$ cd ~/catkin_ws
$ catkin_make
$ source devel/setup.bash
$ roscore
$ rosrun learning_topic person_subscriber
$ rosrun learning_topic person_publisher
```

```
hcx@hcx-vpc:~/catkin_ws$ rosrun learning_topic person_subscriber
[ INFO] [1562216316.857673702]: Subcribe Person Info: name:Tom  age:18  sex:1
[ INFO] [1562216317.857324485]: Subcribe Person Info: name:Tom  age:18  sex:1
[ INFO] [1562216318.857310636]: Subcribe Person Info: name:Tom  age:18  sex:1
[ INFO] [1562216319.856921435]: Subcribe Person Info: name:Tom  age:18  sex:1
[ INFO] [1562216320.856461694]: Subcribe Person Info: name:Tom  age:18  sex:1
```

```
hcx@hcx-vpc:~/catkin_ws$ rosrun learning_topic person_publisher
[ INFO] [1562216315.855698333]: Publish Person Info: name:Tom  age:18  sex:1
[ INFO] [1562216316.856484874]: Publish Person Info: name:Tom  age:18  sex:1
[ INFO] [1562216317.856251972]: Publish Person Info: name:Tom  age:18  sex:1
[ INFO] [1562216318.856513919]: Publish Person Info: name:Tom  age:18  sex:1
[ INFO] [1562216319.856089664]: Publish Person Info: name:Tom  age:18  sex:1
[ INFO] [1562216320.855924037]: Publish Person Info: name:Tom  age:18  sex:1
```

创建发布者和订阅者代码（Python）

古月居 GYH.AI

person_publisher.py

```python
#!/usr/bin/env python
# -*- coding: utf-8 -*-
# 该例程将发布/person_info话题，自定义消息类型learning_topic::Person

import rospy
from learning_topic.msg import Person

def velocity_publisher():
    # ROS节点初始化
    rospy.init_node('person_publisher', anonymous=True)

    # 创建一个Publisher，发布名为/person_info的topic，消息类型为learning_topic::Person，队列长度10
    person_info_pub = rospy.Publisher('/person_info', Person, queue_size=10)

    #设置循环的频率
    rate = rospy.Rate(10)

    while not rospy.is_shutdown():
        # 初始化learning_topic::Person类型的消息
        person_msg = Person()
        person_msg.name = "Tom";
        person_msg.age  = 18;
        person_msg.sex  = Person.male;

        # 发布消息
        person_info_pub.publish(person_msg)
        rospy.loginfo("Publsh person message[%s, %d, %d]",
                person_msg.name, person_msg.age, person_msg.sex)

        # 按照循环频率延时
        rate.sleep()

if __name__ == '__main__':
    try:
        velocity_publisher()
    except rospy.ROSInterruptException:
        pass
```

person_subscriber.py

```python
#!/usr/bin/env python
# -*- coding: utf-8 -*-
# 该例程将订阅/person_info话题，自定义消息类型learning_topic::Person

import rospy
from learning_topic.msg import Person

def personInfoCallback(msg):
    rospy.loginfo("Subcribe Person Info: name:%s  age:%d  sex:%d",
             msg.name, msg.age, msg.sex)

def person_subscriber():
    # ROS节点初始化
    rospy.init_node('person_subscriber', anonymous=True)

    # 创建一个Subscriber，订阅名为/person_info的topic，注册回调函数personInfoCallback
    rospy.Subscriber("/person_info", Person, personInfoCallback)

    # 循环等待回调函数
    rospy.spin()

if __name__ == '__main__':
    person_subscriber()
```

感谢观看

怕什么真理无穷，进一寸有一寸的欢喜

更多精彩，欢迎关注

古月居

古月学院

ROS入门
21讲

古月居 搜一下